AMENDMENTS

In the Specification

Please insert the following paragraph before paragraph 0001 and before the heading "Field of the Invention."

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

Please amended paragraph 0037 as follows:

[0037] X represents the [[a]] partitioning factor that, in one embodiment, is based on the geometric location of DUT 511 in the test structure 501. Referring back to Figure 5, in one embodiment, X is calculated as the distance from input port 503 to output port 507 (a+b) divided by the distance from DUT 511 to input port 503 (a). In embodiments where a is equal to b, the X partitioning factor is 2.

Please amend the specification beginning on page 7, paragraph 0046 to top of page 11 before paragraph 0049 as follows to increase the spacing between the lines of the computer code:

[0046] Below is listed one embodiment of code for partitioning the measured S parameters of a thru test structure (e.g. 601) into a set of S parameters representative of an input network (e.g. 809).

```
i1=0
 i=0
 WHILE i < SIZE(thru)
   In_2_port.11[i]=thru.11[i1]
   !
   !********* S12 *********
   !

Rev_Mag.12[i]=(real(thru.12[i1])^2+imag(thru.12[i1])^2)^0.5
   Mag_In_2_port.12[i]=Rev_Mag.12[i]^var2
   !
```

```
   Rev_Arg.12[i]=imag(thru.12[i1])//real(thru.12[i1])
   Angle_In_2_port.12[i]=atn(Rev_Arg.12[i])//var1
   !
   ! if Angle_In_2_port.12[i] > 0 then
   !    Angle_In_2_port.12[i] = Angle_In_2_port.12[i]-1.57
   ! end if
   !

Real_In_2_port.12[i]=cos(pct_1*Angle_In_2_port.12[i])*Mag_I
n_2_port.12[i]

Imag_In_2_port.12[i]=sin(pct_1*Angle_In_2_port.12[i])*Mag_I
n_2_port.12[i]
   !

In_2_port.12[i]=(Real_In_2_port.12[i])+j*(Imag_In_2_port.12
[i])
   !
   !********* S21 *********
   !

Fwd_Mag.21[i]=(real(thru.21[i1])^2+imag(thru.21[i1])^2)^0.5
   Mag_In_2_port.21[i]=Fwd_Mag.21[i]^var2
   !
   Fwd_Arg.21[i]=imag(thru.21[i1])//real(thru.21[i1])
   Angle_In_2_port.21[i]=atn(Fwd_Arg.21[i])//var1
   !
   ! if Angle_In_2_port.21[i] > 0 then
   !    Angle_In_2_port.21[i] = Angle_In_2_port.21[i]-1.57
   ! end if
   !
```

```
Real_In_2_port.21[i]=cos(pct_1*Angle_In_2_port.21[i])*Mag_I
n_2_port.21[i]

Imag_In_2_port.21[i]=sin(pct_1*Angle_In_2_port.21[i])*Mag_I
n_2_port.21[i]
   !

In_2_port.21[i]=(Real_In_2_port.21[i])+j*(Imag_In_2_port.21
[i])
   !
   In_2_port.22[i]=0+j0
   i1=i1+1
   IF i1>=size_thru THEN i1=0
   i = i + 1
 END WHILE
 !
 RETURN In_2_port
```

[0047] Below is listed one embodiment of code for partitioning the S parameters of a thru test structure (e.g. 601) into a set of S parameters representative of an output network (e.g. 811).

```
i1=0
 i=0
 WHILE i < SIZE(thru)
   Out_2_port.11[i]=0+j0
   !
   !********* S12 ***********
   !

Rev_Mag.12[i]=(real(thru.12[i1])^2+imag(thru.12[i1])^2)^0.5
   Mag_Out_2_port.12[i]=(Rev_Mag.12[i])^var2
```

```
    !
    Rev_Arg.12[i]=imag(thru.12[i1])//real(thru.12[i1])
    Angle_Out_2_port.12[i]=atn(Rev_Arg.12[i])//var1
    !
    ! if Angle_Out_2_port.12[i] > 0 then
    !    Angle_Out_2_port.12[i] = Angle_Out_2_port.12[i]-
1.57
    ! end if
    !

Real_Out_2_port.12[i]=cos(pct_1*Angle_Out_2_port.12[i])*Mag
_Out_2_port.12[i]

Imag_Out_2_port.12[i]=sin(pct_1*Angle_Out_2_port.12[i])*Mag
_Out_2_port.12[i]
    !

Out_2_port.12[i]=(Real_Out_2_port.12[i])+j*(Imag_Out_2_port
.12[i])
    !
    !********* S21 ***********
    !

Fwd_Mag.21[i]=(real(thru.21[i1])^2+imag(thru.21[i1])^2)^0.5
    Mag_Out_2_port.21[i]=(Fwd_Mag.21[i])^var2
    !
    Fwd_Arg.21[i]=imag(thru.21[i1])//real(thru.21[i1])
    Angle_Out_2_port.21[i]=atn(Fwd_Arg.21[i])//var1
    !
    ! if Angle_Out_2_port.21[i] > 0 then
    !    Angle_Out_2_port.21[i] = Angle_Out_2_port.21[i]-
1.57
    ! end if
```

```
    !
Real_Out_2_port.21[i]=cos(pct_1*Angle_Out_2_port.21[i])*Mag
_Out_2_port.21[i]

Imag_Out_2_port.21[i]=sin(pct_1*Angle_Out_2_port.21[i])*Mag
_Out_2_port.21[i]
    !

Out_2_port.21[i]=(Real_Out_2_port.21[i])+j*(Imag_Out_2_port
.21[i])
    !
    Out_2_port.22[i]=thru.22[i1]
    i1=i1+1
    IF i1>=size_thru THEN i1=0
    i = i + 1
  END WHILE
  !
  RETURN Out_2_port
```

[0048] Below is one embodiment of code for converting a set of measured DUT test structure S parameters and sets of S parameters representative of an input network and an output network to sets of ABCD parameters. The code below also obtains the intrinsic characteristics of the DUT in ABCD parameters and converts them back to S parameters.

```
i1=0
 i=0
 WHILE i < SIZE(total)
   L_dummy_act.11[i]=L_dummy.11[i1]
   L_dummy_act.12[i]=L_dummy.12[i1]
   L_dummy_act.21[i]=L_dummy.21[i1]
   L_dummy_act.22[i]=L_dummy.22[i1]
   R_dummy_act.11[i]=R_dummy.11[i1]
```

```
    R_dummy_act.12[i]=R_dummy.12[i1]
    R_dummy_act.21[i]=R_dummy.21[i1]
    R_dummy_act.22[i]=R_dummy.22[i1]
    i1=i1+1
    IF i1>=size_dummy THEN i1=0
    i = i + 1
  END WHILE
  !
  PRINT "now do the de-embedding using ABCD matrix
manipulation ..."
  !
  xtor=TwoPort(L_dummy_act,"S","A")^-1 *
TwoPort(total,"S","A") * TwoPort(R_dummy_act,"S","A")^-1
  xtor=TwoPort(xtor,"A","S")
  !
  RETURN xtor
```